

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: data=pd.read_csv('/content/Titanic-Dataset.csv')
data.head()
```

```
Out[2]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	I
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	I
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	I

In [3]: data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     891 non-null    int64
1   Survived        891 non-null    int64
2   Pclass         891 non-null    int64
3   Name            891 non-null    object
4   Sex             891 non-null    object
5   Age             714 non-null    float64
6   SibSp           891 non-null    int64
7   Parch           891 non-null    int64
8   Ticket          891 non-null    object
9   Fare            891 non-null    float64
10  Cabin           204 non-null    object
11  Embarked        889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [4]: data.describe()

Out[4]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
In [5]: corr=data.corr()
corr
```

<ipython-input-5-0d3ae1d0be10>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

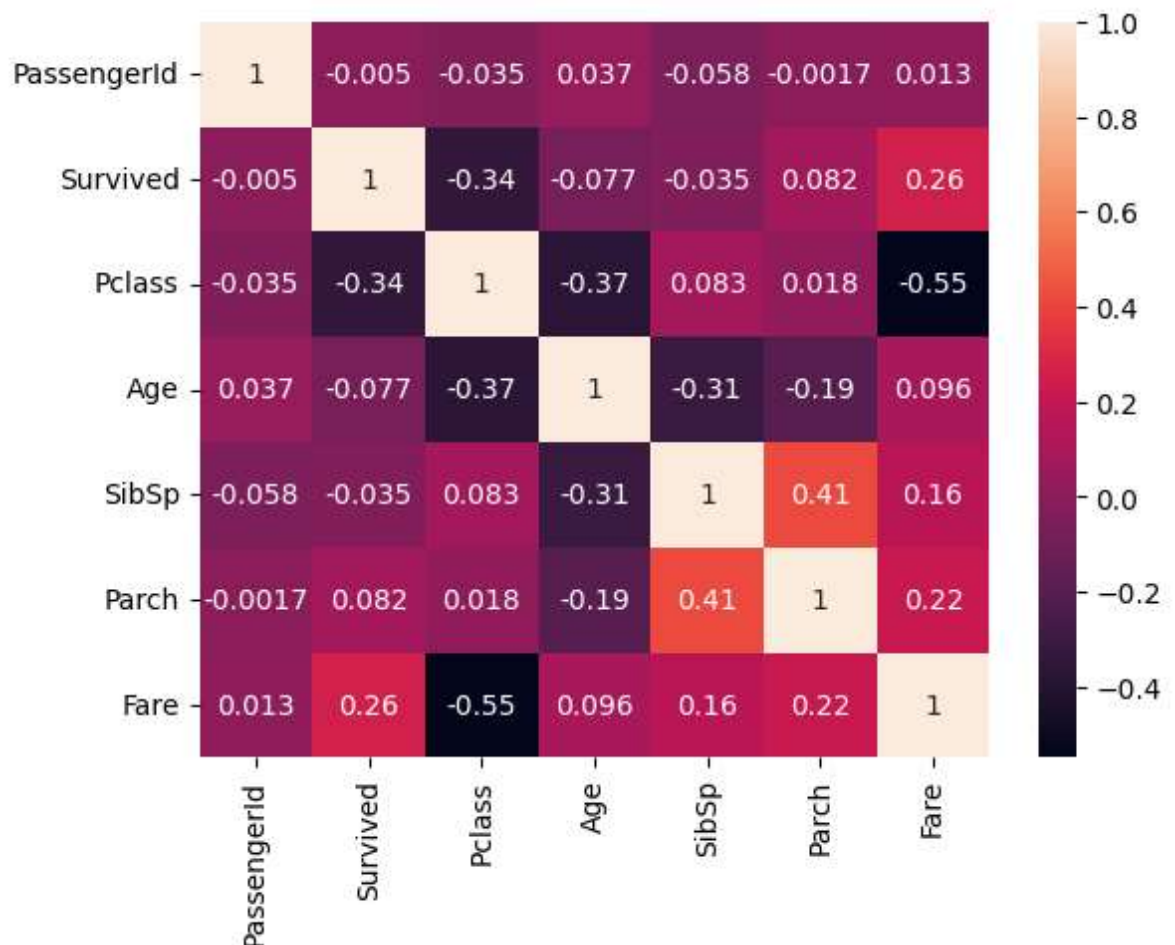
```
corr=data.corr()
```

```
Out[5]:
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

```
In [6]: sns.heatmap(corr,annot=True)
```

```
Out[6]: <Axes: >
```



```
In [7]: data.Cabin.value_counts()
```

```
Out[7]: B96 B98      4
        G6         4
        C23 C25 C27  4
        C22 C26      3
        F33         3
        ..
        E34         1
        C7          1
        C54         1
        E36         1
        C148        1
        Name: Cabin, Length: 147, dtype: int64
```

```
In [8]: data.Embarked.value_counts()
```

```
Out[8]: S      644
        C      168
        Q       77
        Name: Embarked, dtype: int64
```

```
In [9]: data.Parch.value_counts()
```

```
Out[9]: 0      678
        1      118
        2       80
        5        5
        3        5
        4        4
        6        1
        Name: Parch, dtype: int64
```

```
In [10]: data.isnull().any()
```

```
Out[10]: PassengerId  False
         Survived     False
         Pclass       False
         Name         False
         Sex          False
         Age          True
         SibSp        False
         Parch        False
         Ticket       False
         Fare         False
         Cabin        True
         Embarked     True
         dtype: bool
```

```
In [11]: data.isnull().sum()
```

```
Out[11]: PassengerId      0
         Survived        0
         Pclass          0
         Name            0
         Sex             0
         Age            177
         SibSp           0
         Parch           0
         Ticket          0
         Fare            0
         Cabin          687
         Embarked        2
         dtype: int64
```

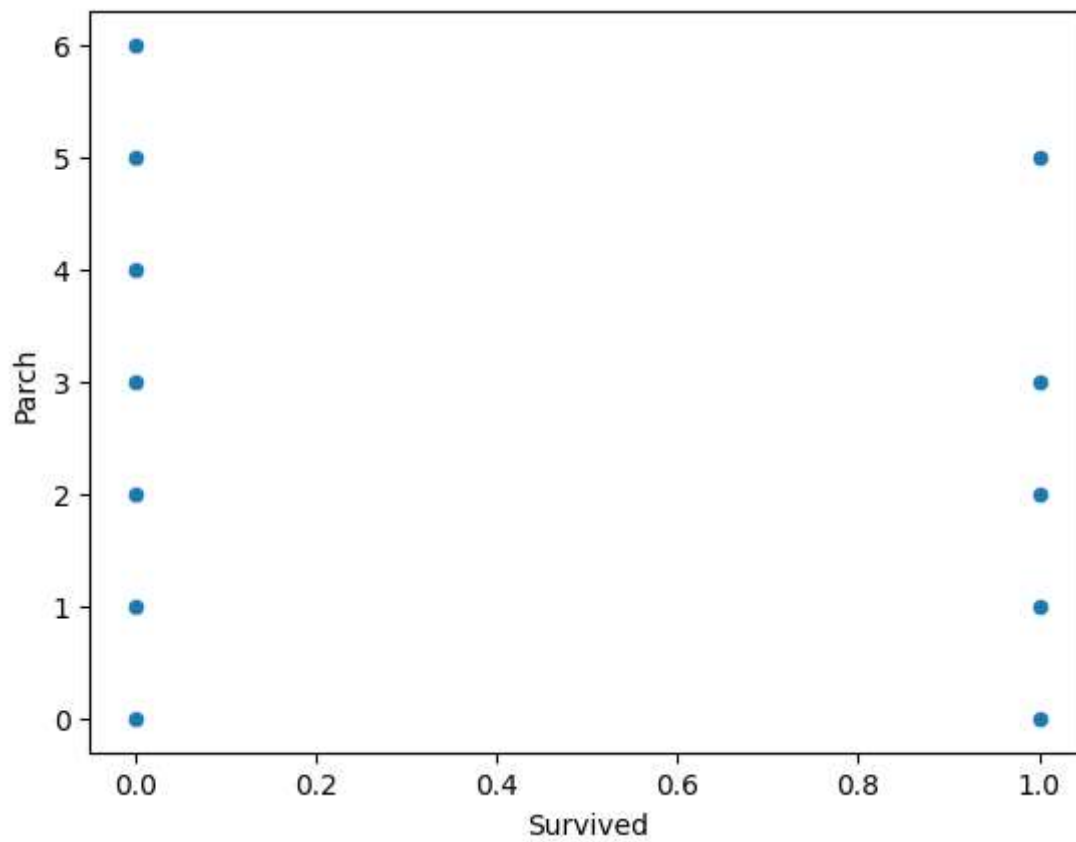
```
In [12]: data["Age"].fillna(data["Age"].mean(),inplace=True)
         data["Cabin"].fillna(data["Cabin"].mode()[0],inplace=True)
         data["Embarked"].fillna(data["Embarked"].mode()[0],inplace=True)
```

```
In [13]: data.isnull().sum()#I removed all null values
```

```
Out[13]: PassengerId      0
         Survived        0
         Pclass          0
         Name            0
         Sex             0
         Age             0
         SibSp           0
         Parch           0
         Ticket          0
         Fare            0
         Cabin            0
         Embarked        0
         dtype: int64
```

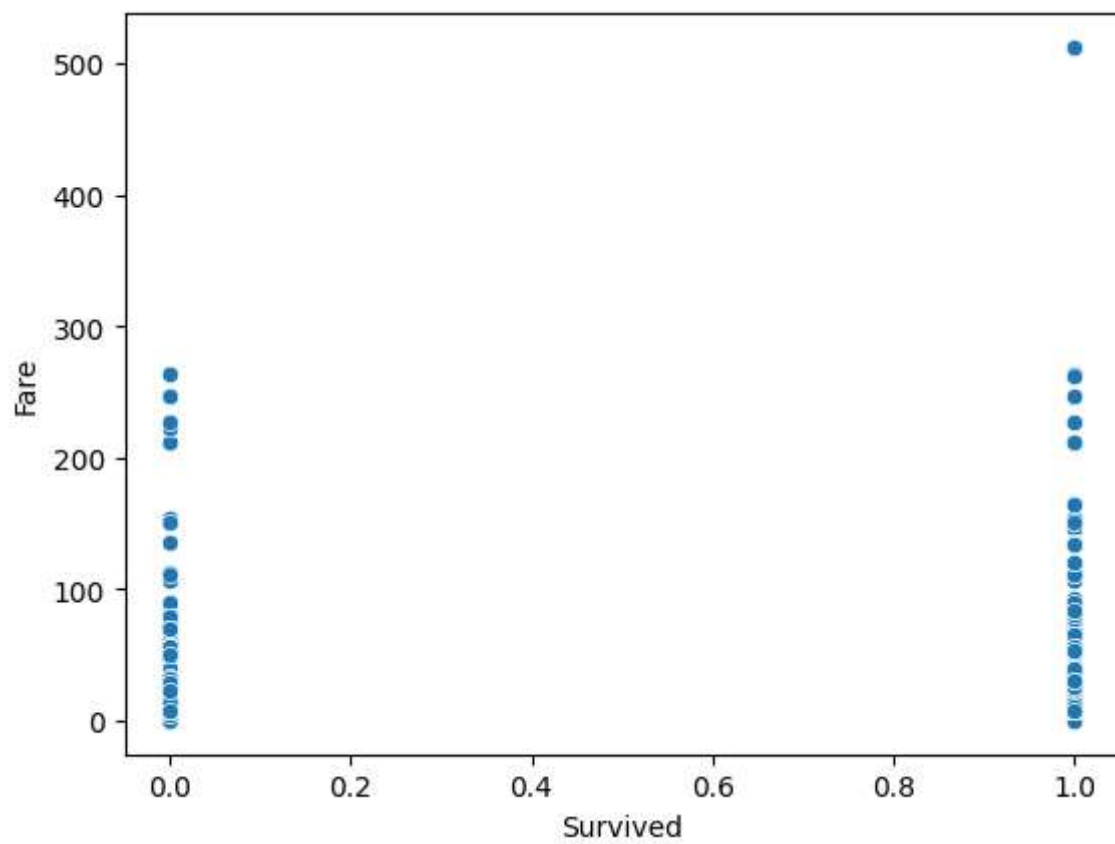
```
In [14]: sns.scatterplot(x=data["Survived"],y=data["Parch"])
```

```
Out[14]: <Axes: xlabel='Survived', ylabel='Parch'>
```



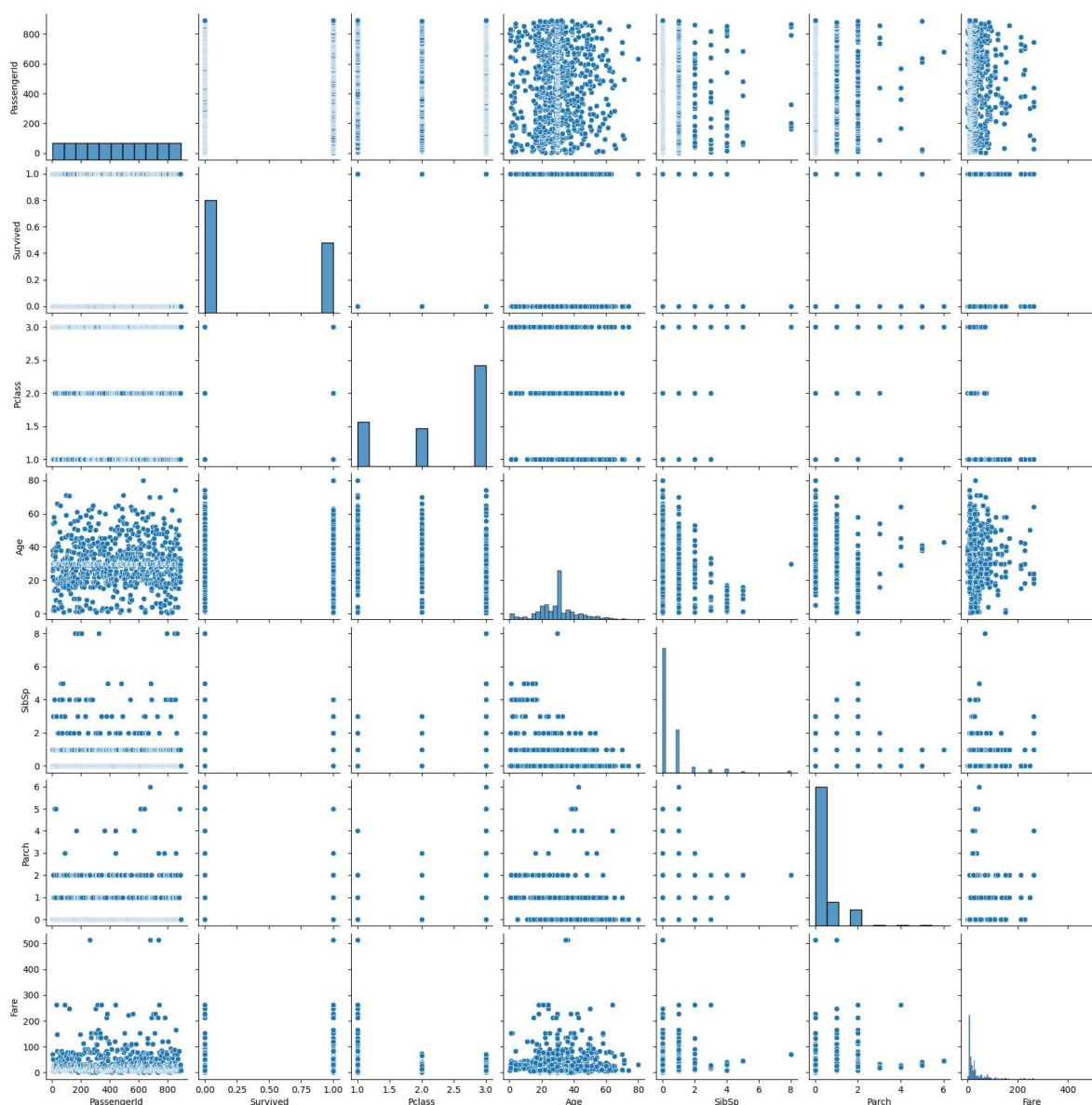
```
In [15]: sns.scatterplot(x=data["Survived"],y=data["Fare"])
```

```
Out[15]: <Axes: xlabel='Survived', ylabel='Fare'>
```



```
In [16]: sns.pairplot(data)
```

```
Out[16]: <seaborn.axisgrid.PairGrid at 0x7b14cb8ebd60>
```



```
In [17]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
In [18]: data["Sex"]=le.fit_transform(data["Sex"])
```

```
In [19]: data["Embarked"]=le.fit_transform(data["Embarked"])
```

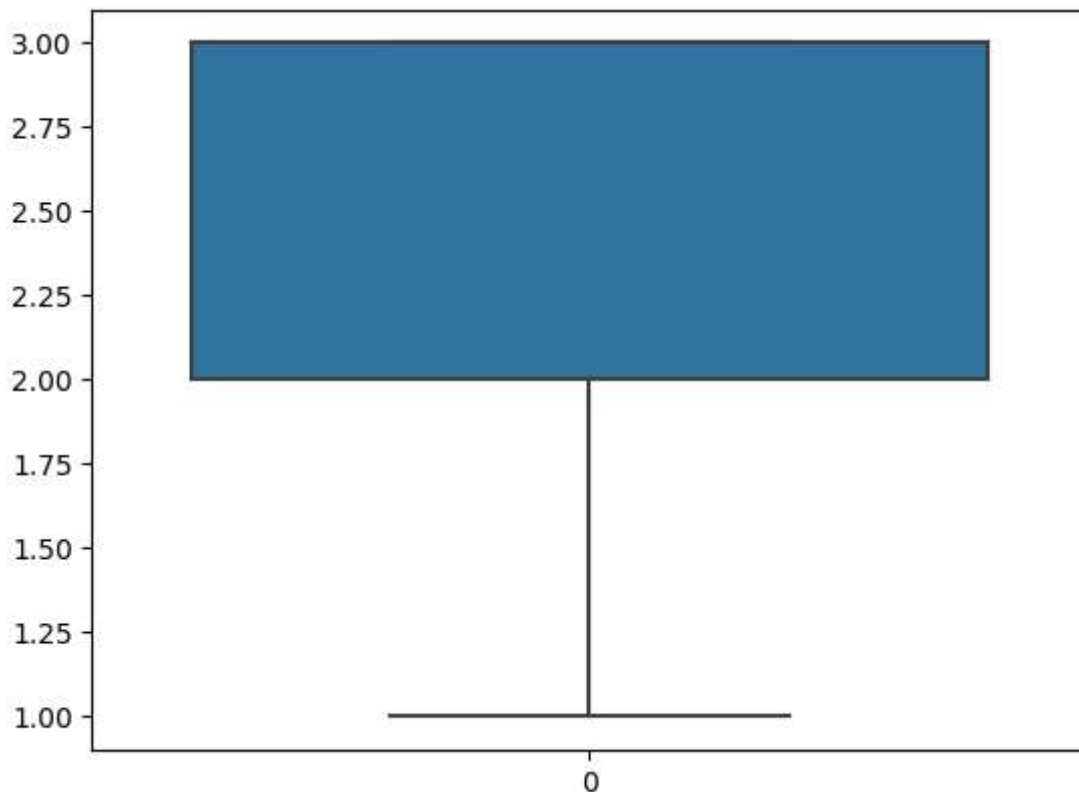

In [20]: `data.head()`

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	B9 B9
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	B9 B9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	B9 B9

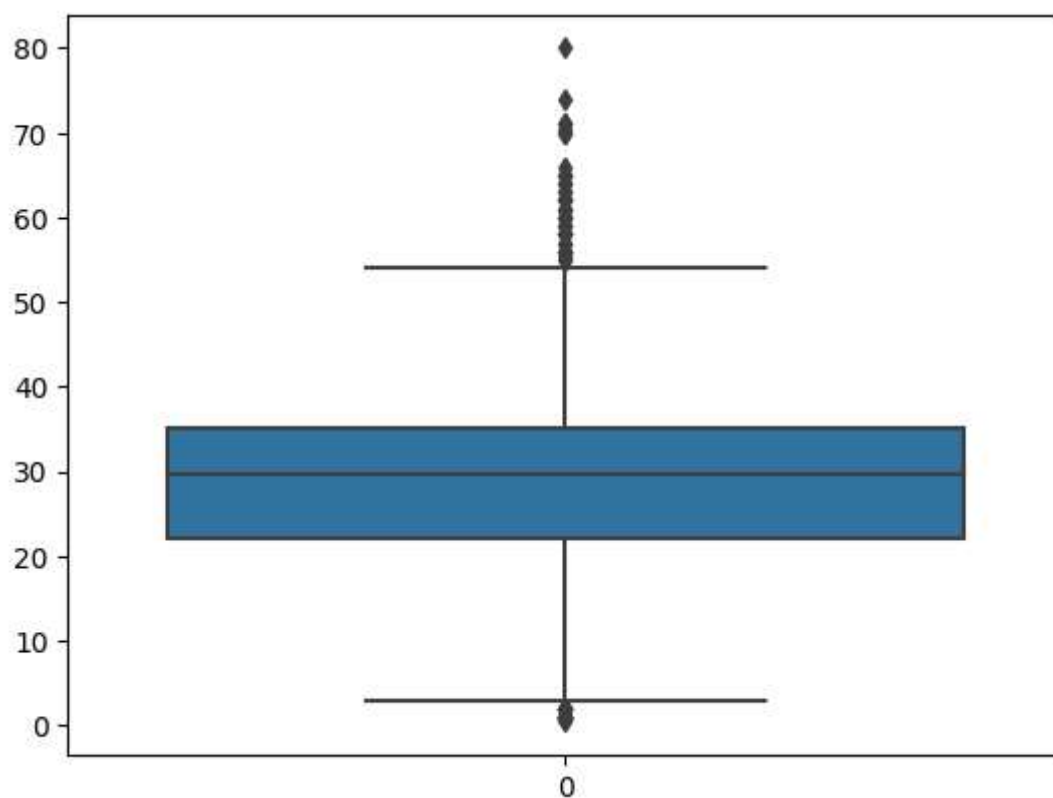
In [21]: `sns.boxplot(data['Pclass'])`

Out[21]: <Axes: >



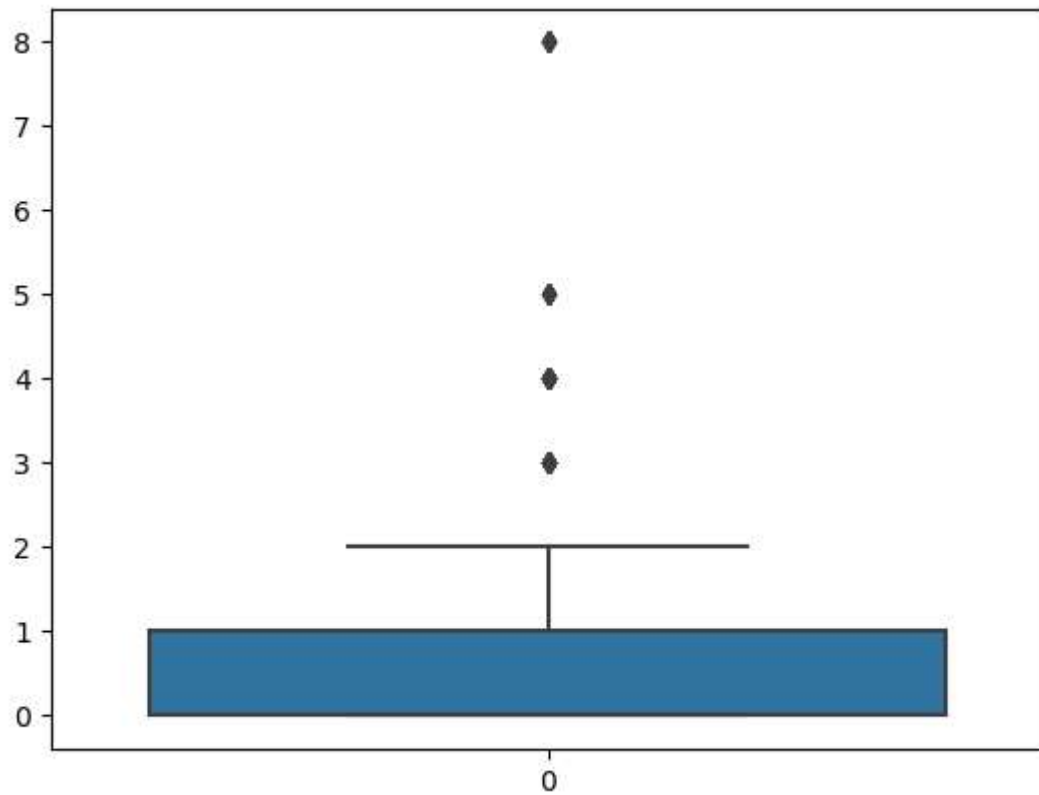
```
In [22]: sns.boxplot(data['Age'])
```

```
Out[22]: <Axes: >
```



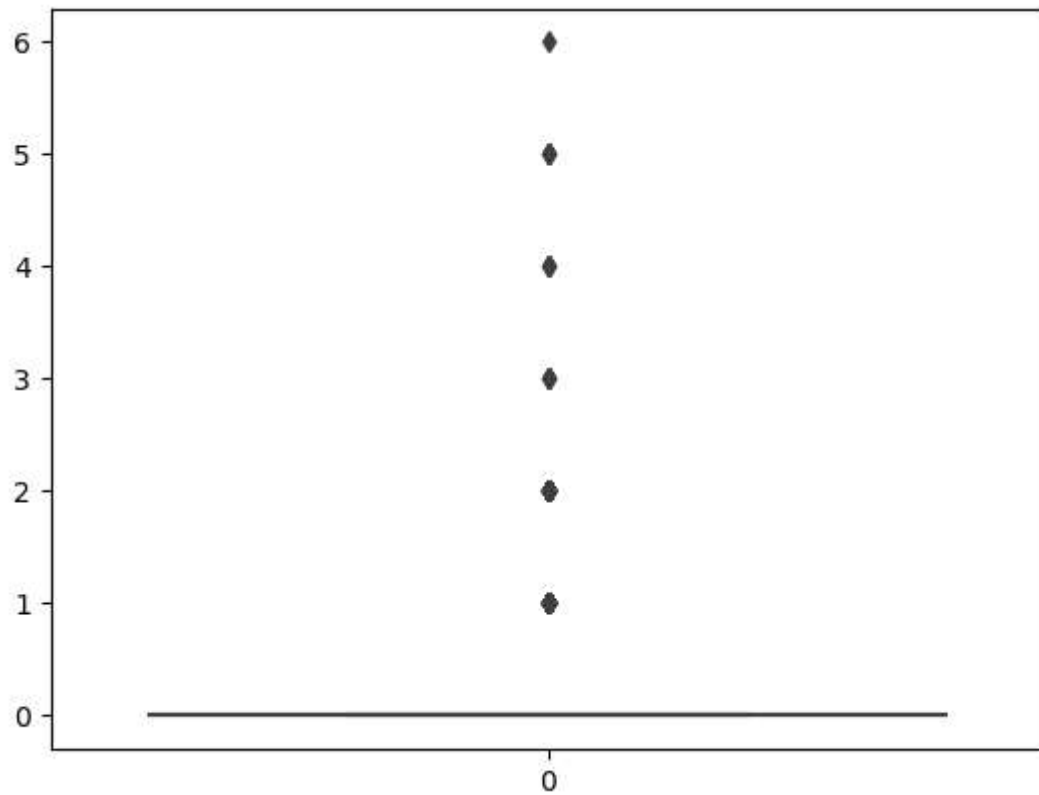
```
sns.boxplot(data['SibSp'])
```

```
Out[23]: <Axes: >
```



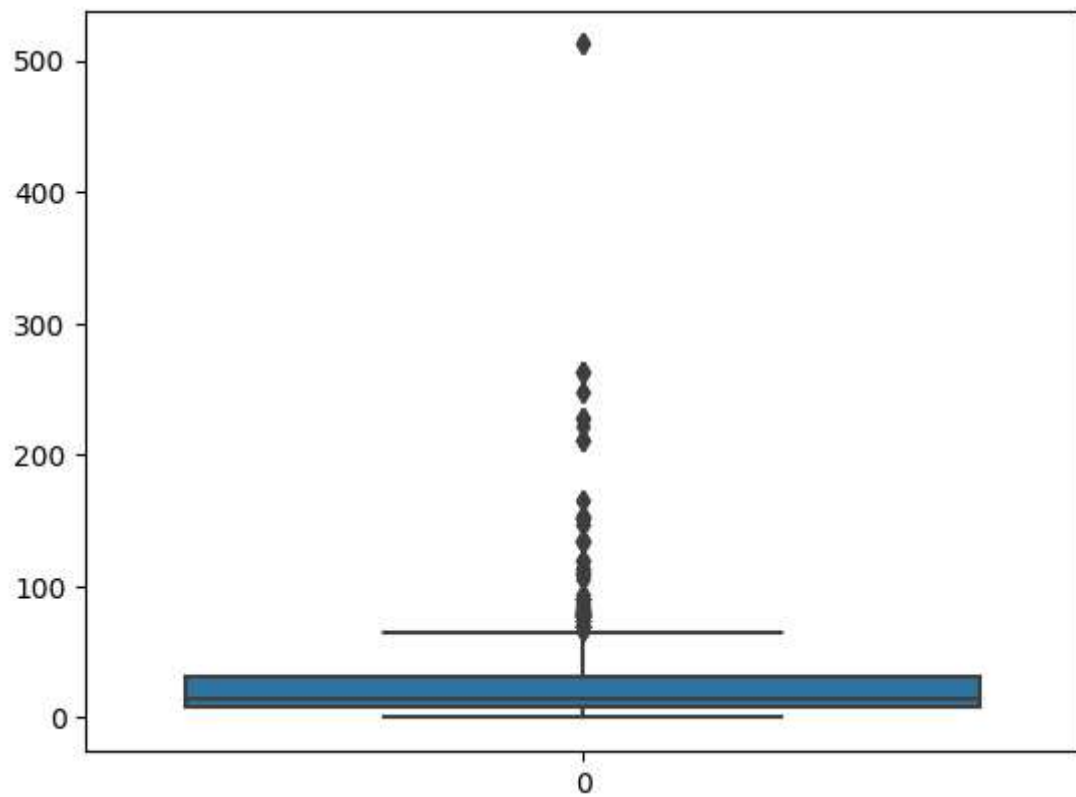
```
In [24]: sns.boxplot(data['Parch'])
```

```
Out[24]: <Axes: >
```



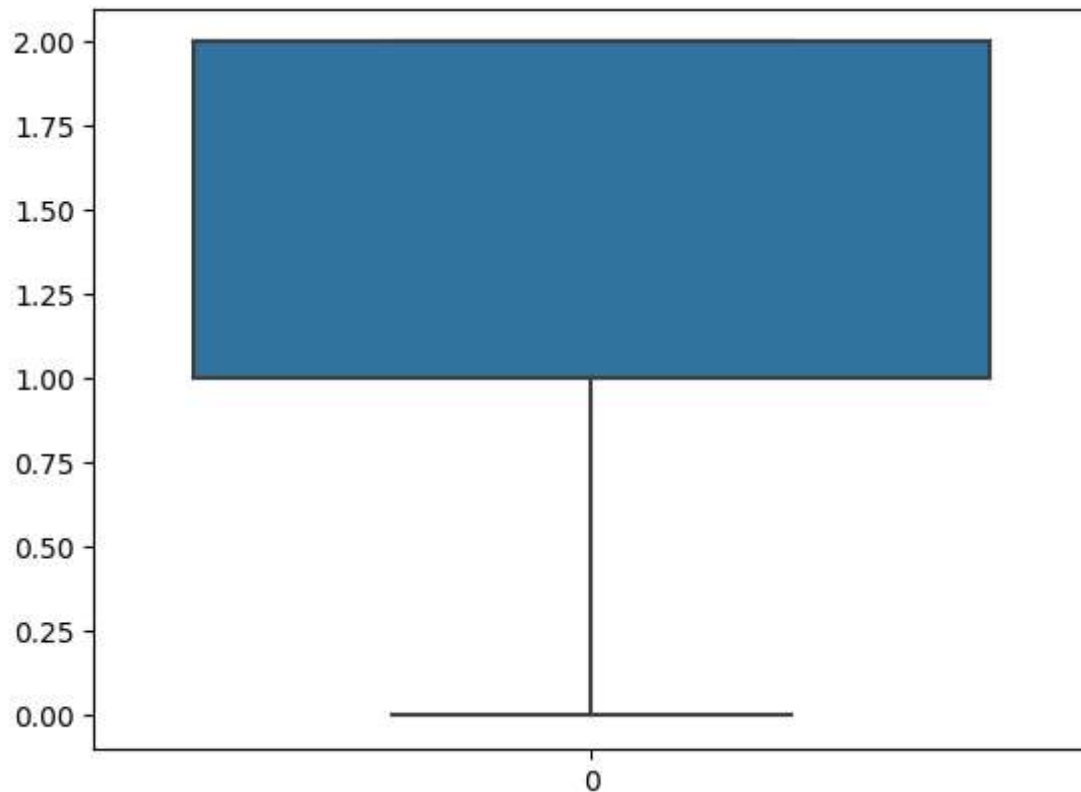
```
In [25]: sns.boxplot(data['Fare'])
```

```
Out[25]: <Axes: >
```



```
In [26]: sns.boxplot(data['Embarked'])
```

```
Out[26]: <Axes: >
```



```
In [27]: q1=data.Age.quantile(0.25)
q3=data.Age.quantile(0.75)
print(q1)
print(q3)
```

```
22.0
35.0
```

```
In [28]: iqr=q3-q1
iqr
```

```
Out[28]: 13.0
```

```
In [29]: upperlimit = q3+1.5*iqr
upperlimit
```

```
Out[29]: 54.5
```

```
In [30]: lowerlimit=q1-1.5*iqr
lowerlimit
```

```
Out[30]: 2.5
```

```
In [31]: data.median()
```

```
<ipython-input-31-135339ac59ce>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
```

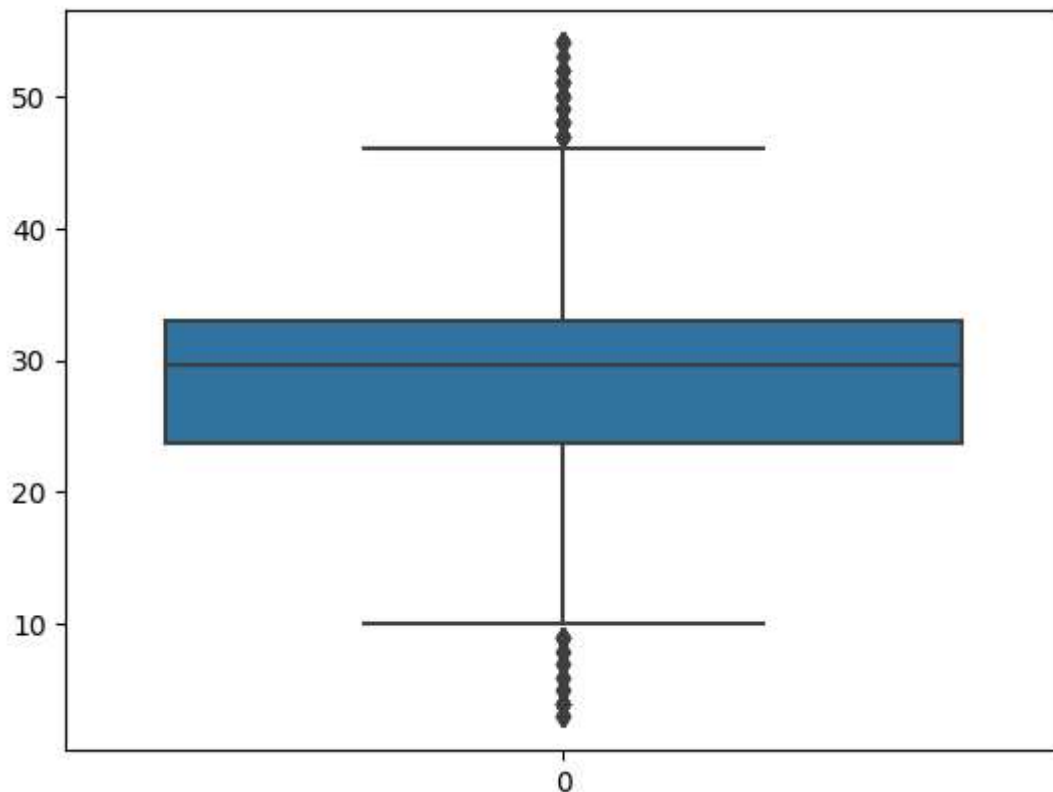
```
data.median()
```

```
Out[31]: PassengerId    446.000000
Survived              0.000000
Pclass               3.000000
Sex                  1.000000
Age                 29.699118
SibSp                0.000000
Parch                0.000000
Fare                14.454200
Embarked            2.000000
dtype: float64
```

```
In [32]: data['Age']=np.where(data['Age']>upperlimit,29.699118,data['Age'])
data['Age'] = np.where(data['Age'] < lowerlimit,29.699118, data['Age'])
```

```
In [33]: sns.boxplot(data['Age'])
```

```
Out[33]: <Axes: >
```



```
In [34]: q1=data.SibSp.quantile(0.25)
         q3=data.SibSp.quantile(0.75)
         print(q1)
         print(q3)
```

```
0.0
1.0
```

```
In [35]: iqr=q3-q1
         iqr
```

```
Out[35]: 1.0
```

```
In [36]: upperlimit = q3+1.5*iqr
         upperlimit
```

```
Out[36]: 2.5
```

```
In [37]: lowerlimit=q1-1.5*iqr
         lowerlimit
```

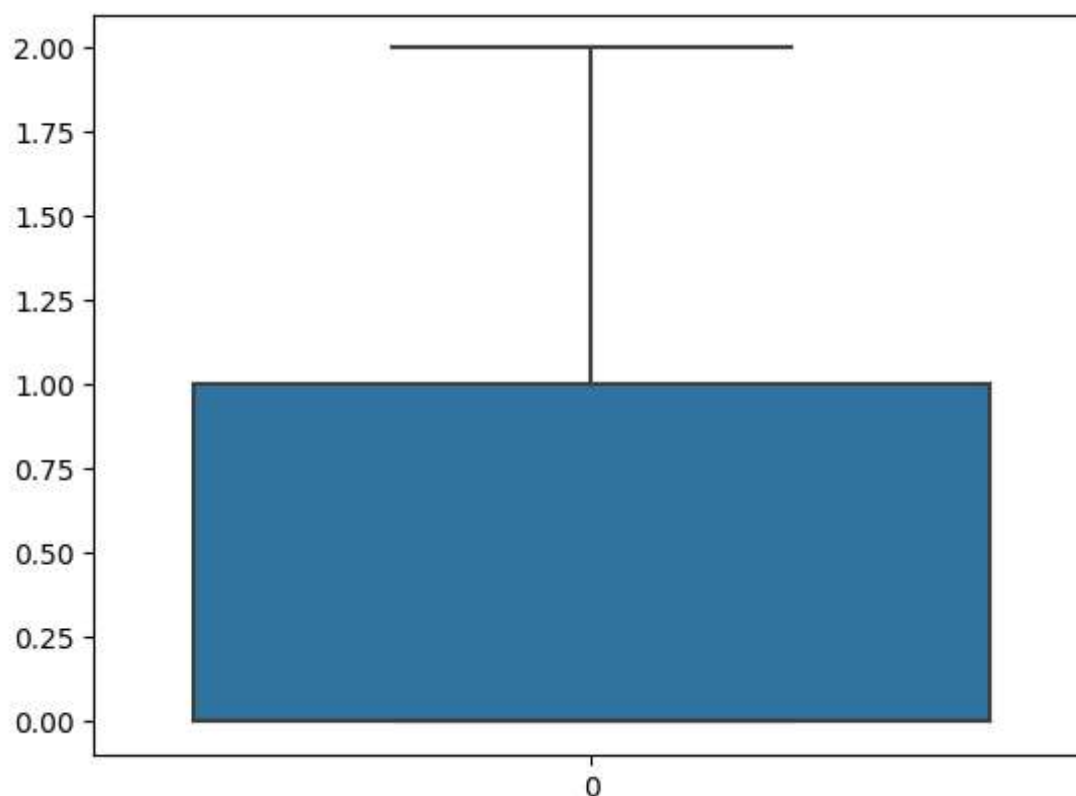
```
Out[37]: -1.5
```

```
In [38]: data['SibSp']=np.where(data['SibSp']>upperlimit,0.000000,data['SibSp'])
```



```
In [39]: sns.boxplot(data['SibSp'])
```

```
Out[39]: <Axes: >
```



```
In [40]: q1=data.Parch.quantile(0.25)
q3=data.Parch.quantile(0.75)
print(q1)
print(q3)
```

```
0.0
0.0
```

```
In [41]: iqr=q3-q1
iqr
```

```
Out[41]: 0.0
```

```
In [ ]: upperlimit = q3+1.5*iqr
upperlimit
```

```
Out[46]: 0.0
```

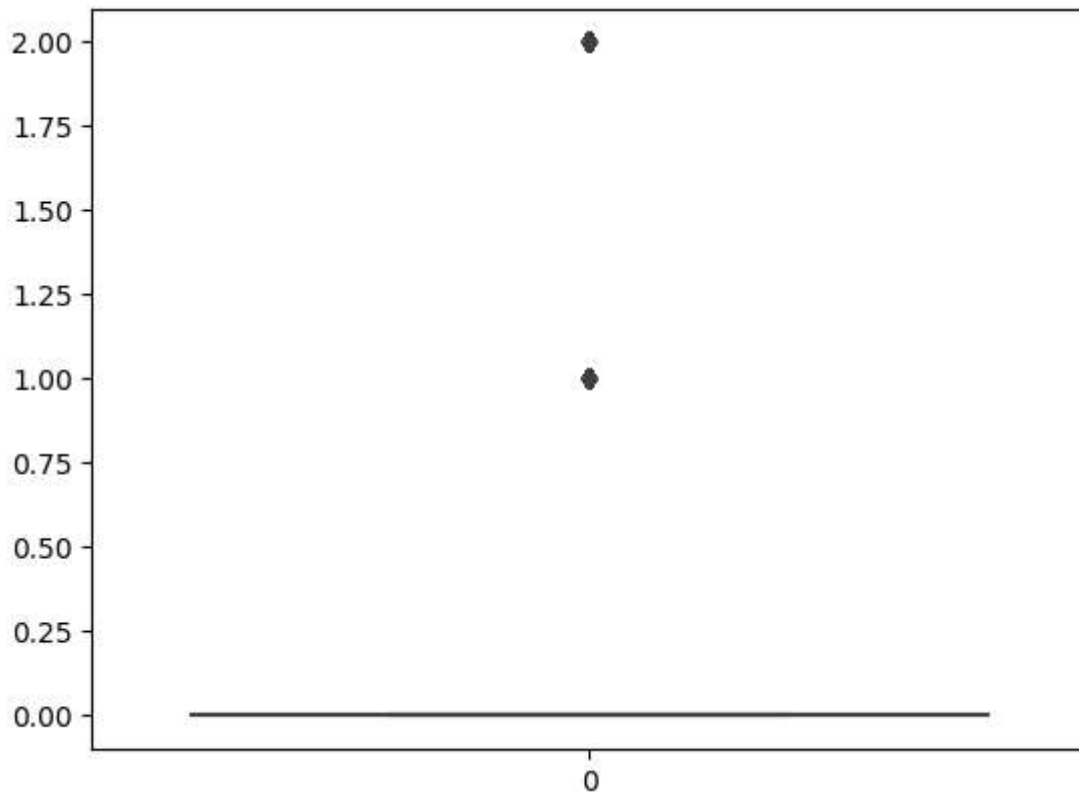
```
In [42]: lowerlimit=q1-1.5*iqr
lowerlimit
```

```
Out[42]: 0.0
```

```
In [43]: data['Parch']=np.where(data['Parch']>upperlimit,0.000000,data['Parch'])
```

```
In [44]: sns.boxplot(data['Parch'])
```

Out[44]: <Axes: >



```
In [45]: q1=data.Fare.quantile(0.25)
q3=data.Fare.quantile(0.75)
print(q1)
print(q3)
```

7.9104

31.0

```
In [46]: iqr=q3-q1
iqr
```

Out[46]: 23.0896

```
In [47]: upperlimit = q3+1.5*iqr
upperlimit
```

Out[47]: 65.6344

```
In [48]: lowerlimit=q1-1.5*iqr  
lowerlimit
```

Out[48]: -26.724

```
In [49]: data.median()
```

<ipython-input-49-135339ac59ce>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

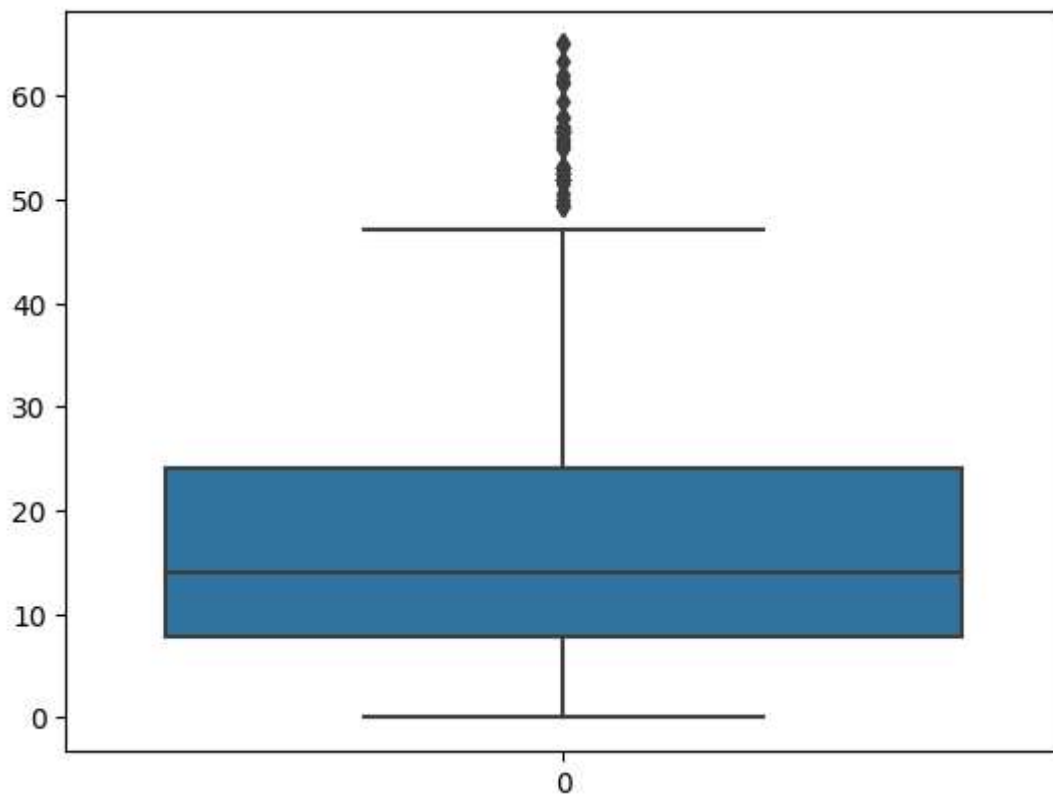
```
data.median()
```

Out[49]: PassengerId 446.000000
Survived 0.000000
Pclass 3.000000
Sex 1.000000
Age 29.699118
SibSp 0.000000
Parch 0.000000
Fare 14.454200
Embarked 2.000000
dtype: float64

```
In [50]: data['Fare']=np.where(data['Fare']>upperlimit,14.054150,data['Fare'])
```

```
In [51]: sns.boxplot(data.Fare)
```

```
Out[51]: <Axes: >
```



```
In [52]: y=data["Survived"]
```

```
In [53]: X=data.drop(columns=["Name","PassengerId","Survived","Ticket","Cabin"],axis=1)
```

```
In [54]: y.head()
```

```
Out[54]: 0    0
         1    1
         2    1
         3    1
         4    0
         Name: Survived, dtype: int64
```

```
In [55]: from sklearn.preprocessing import MinMaxScaler
         ms=MinMaxScaler()
```

```
In [56]: X_Scaled=ms.fit_transform(X)
```

```
In [57]: X_Scaled=pd.DataFrame(ms.fit_transform(X),columns=X.columns)
```

```
In [58]: X_Scaled.head()
```

```
Out[58]:
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1.0	1.0	0.372549	0.5	0.0	0.111538	1.0
1	0.0	0.0	0.686275	0.5	0.0	0.216218	0.0
2	1.0	0.0	0.450980	0.0	0.0	0.121923	1.0
3	0.0	0.0	0.627451	0.5	0.0	0.816923	1.0
4	1.0	1.0	0.627451	0.0	0.0	0.123846	1.0

```
In [59]: from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test = train_test_split(X_Scaled,y,test_size =0.2,ran
```

```
In [60]: print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(712, 7) (179, 7) (712,) (179,)
```